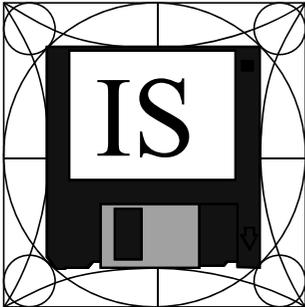




Verifica e validazione: introduzione

Ingegneria del Software
V. Ambriola, G.A. Cignoni,
C. Montangero, L. Semini

Aggiornamenti di: T. Vardanega (UniPD)



Dipartimento di Informatica, Università di Pisa

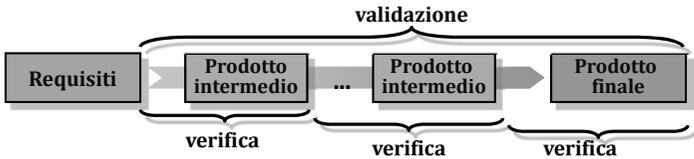
1/22



Verifica e validazione

In pratica ...

- ❑ La verifica accerta che l'esecuzione delle attività di processo attuate nel periodo in esame («fase») non abbia introdotto errori
 - Per questo, la verifica agevola il successo della validazione
- ❑ La validazione accerta che il prodotto realizzato sia pienamente conforme alle attese



```
graph LR; R[Requisiti] --> PI1[Prodotto intermedio]; PI1 -- verifica --> PI2[Prodotto intermedio]; PI2 -- verifica --> PF[Prodotto finale]; subgraph validazione; PI1 -- validazione --> PF; end
```

Dipartimento di Informatica, Università di Pisa

3/22



Verifica e validazione

Verifica e validazione

- ❑ **Software verification**
 - Provides objective evidence that the outputs of a particular phase of the software development life cycle meet all of the specified [process] requirements for that phase
 - Software verification looks for consistency, completeness, and correctness of the [examined product items], and provides support for a subsequent conclusion that software is validated
- ❑ **Software validation**
 - Confirmation by examination and provision of objective evidence that the software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled

Dipartimento di Informatica, Università di Pisa

2/22



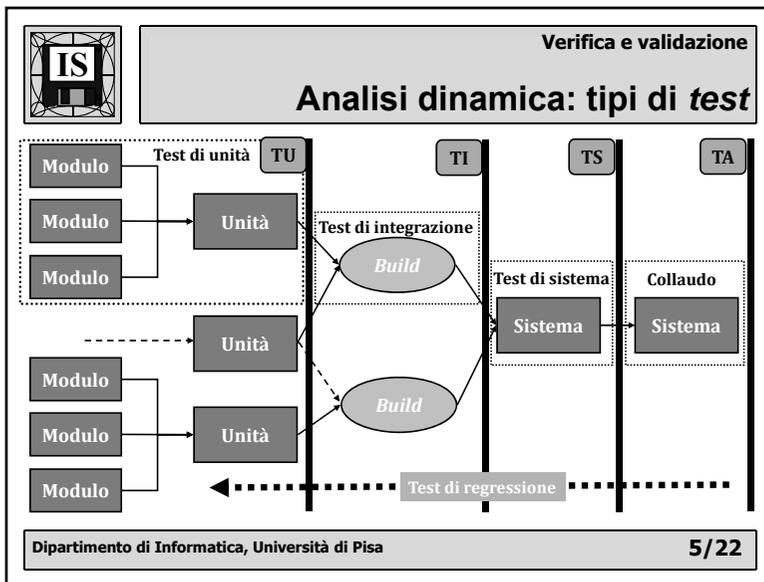
Verifica e validazione

Forme di verifica

- ❑ **Analisi dinamica**
 - Richiede esecuzione del programma
 - Quindi si può effettuare solo dopo l'inizio della codifica
 - Viene effettuata tramite prove (test)
 - Usata sia nella verifica che nella validazione
- ❑ **Analisi statica**
 - Non richiede esecuzione del prodotto SW in alcuna sua parte
 - Quindi si può attuare da subito (meno male!)
 - Studia la documentazione e il codice (sorgente e oggetto)
 - Accerta conformità a regole, assenza di difetti, presenza di proprietà desiderate

Dipartimento di Informatica, Università di Pisa

4/22



Verifica e validazione

Analisi dinamica: ambiente di prova

- I test devono essere ripetibili: per questo specificano
 - Ambiente d'esecuzione: HW/SW, stato iniziale
 - Attese: ingressi richiesti, uscite ed effetti attesi
 - Procedure: esecuzione, analisi dei risultati
- I test vanno automatizzati: per questo usano strumenti
 - Driver componente attiva fittizia per pilotare il test
 - Stub componente passiva fittizia per simulare la parte del sistema utile al test ma non oggetto di esso (vedere alla voce: «Per approfondire»)
 - Logger componente non intrusivo di registrazione dei dati di esecuzione per analisi dei risultati

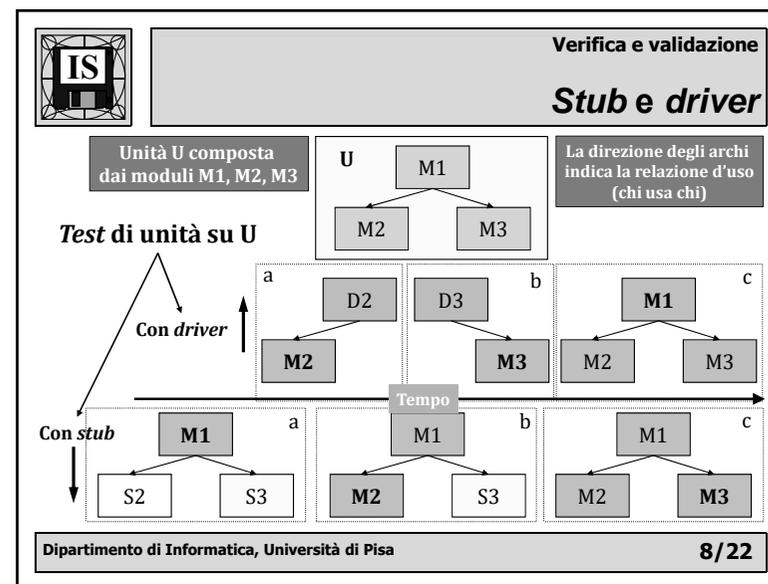
Dipartimento di Informatica, Università di Pisa 7/22

Verifica e validazione

Glossario

- Unità
 - La più piccola quantità di SW che sia verificabile singolarmente
 - Tipicamente prodotta da un singolo programmatore
 - Va sempre intesa in senso architeturale: non linee di codice ma entità di organizzazione logica
 - Singola procedura, singola classe, piccolo aggregato (*package*)
- Il modulo (come determinato dal linguaggio di programmazione) è una frazione dell'unità
- Il componente integra più unità correlate e coese

Dipartimento di Informatica, Università di Pisa 6/22





Verifica e validazione

Test di unità

- ❑ È attività di analisi dinamica
 - È agevolata da attività mirate di analisi statica
 - Per determinare limiti di iterazioni, flusso di programma, valori di variabili, ecc.
 - Si svolge con il massimo grado di parallelismo
 - Poiché i TU sono tanti, conviene massimamente automatizzarli
- ❑ Responsabilità
 - Dello stesso programmatore per le unità più semplici
 - Di un verificatore indipendente altrimenti
- ❑ Obiettivi
 - Verificare la correttezza del codice *as implemented*

Dipartimento di Informatica, Università di Pisa

9/22



Verifica e validazione

Test di regressione

- ❑ Modifiche effettuate per aggiunta, correzione o rimozione, non devono pregiudicare le funzionalità già verificate, causando regressione
 - Tale rischio aumenta all'aumentare dell'accoppiamento e al diminuire dell'incapsulazione
- ❑ Il *test di regressione* comprende tutti i *test* necessari ad accertare che la modifica di una parte P di S non causi errori in P, in S, o in ogni altra parte del sistema che sia in relazione con S
 - Nel caso migliore, si effettua ripetendo *test* già eseguiti

Dipartimento di Informatica, Università di Pisa

11/22



Verifica e validazione

La risoluzione dei problemi

- ❑ La verifica serve per scovare problemi e risolverli tempestivamente
- ❑ La soluzione dei problemi attiene al processo di supporto «*problem resolution*» di ISO 122017
- ❑ L'obiettivo di quel processo è assicurare che tutti i problemi rilevati siano registrati, analizzati, compresi, e risolti
 - Sviluppare una strategia di gestione dei problemi
 - Registrare ogni problema rilevato e classificarlo in uno storico
 - Analizzare ogni problema e determinare soluzioni accettabili
 - Realizzare la soluzione scelta
 - Verificare l'esito della correzione (vedi: Test di Regressione)
 - Assicurare che tutti i problemi noti siano sotto gestione

Dipartimento di Informatica, Università di Pisa

10/22



Verifica e validazione

Test di integrazione

- ❑ Per costruzione e verifica incrementale del sistema
 - Componenti sviluppati in parallelo e verificati incrementalmente
 - La *build* incrementale è automatizzabile
 - In condizioni ottimali l'integrazione è priva di problemi
- ❑ Quali problemi rileva
 - Errori residui nella realizzazione dei componenti
 - Modifica delle interfacce o cambiamenti nei requisiti
 - Riutilizzo di componenti dal comportamento oscuro o inadatto
 - Integrazione con altre applicazioni non ben conosciute

Dipartimento di Informatica, Università di Pisa

12/22



Verifica e validazione
Test di sistema e collaudo

- ❑ **Validazione**
 - **Test di sistema come attività interna del fornitore**
 - Per accertare la copertura dei requisiti SW
 - **Collaudo come attività supervisionata dal committente**
 - Per dimostrazione di conformità del prodotto sulla base di casi di prova specificati nel o implicati dal contratto
- ❑ **Implicazioni contrattuali**
 - Il collaudo è attività formale
 - Al collaudo segue il rilascio del prodotto (con eventuale garanzia) e la fine della commessa (con eventuale manutenzione)

Dipartimento di Informatica, Università di Pisa13/22



Verifica e validazione
Metodi di lettura

- ❑ **Walkthrough e Inspection**
- ❑ **Metodi pratici**
 - Basati su lettura dell'oggetto di verifica
 - Efficacia dipendente dall'esperienza dei verificatori
 - Nell'organizzare le attività di verifica
 - Nel documentare le attività svolte e i risultati ottenuti
- ❑ **Modalità relativamente complementari**

Dipartimento di Informatica, Università di Pisa15/22



Verifica e validazione
Analisi statica

- ❑ Non richiedono esecuzione di parti del sistema SW
- ❑ Applicano a ogni prodotto di processo (non solo SW)
 - Per tutti i processi attivati nel progetto
- ❑ **Metodi di lettura (desk check)**
 - Impiegati solo per prodotti semplici
- ❑ **Metodi formali**
 - Basati sulla prova assistita di proprietà
 - La cui dimostrazione dinamica può essere eccessivamente onerosa
 - Verifica di equivalenza o generazione automatica

Dipartimento di Informatica, Università di Pisa14/22



Verifica e validazione
Walkthrough: definizione

- ❑ **Obiettivo**
 - Rivelare la presenza di difetti
 - Eseguire una lettura critica [del prodotto in esame]
 - A largo spettro
 - Senza l'assunzione di presupposti
- ❑ **Agenti**
 - Gruppi misti ispettori/sviluppatori ma con ruoli ben distinti
- ❑ **Strategia (per il codice)**
 - Percorrerlo simulandone possibili esecuzioni

Dipartimento di Informatica, Università di Pisa16/22



Verifica e validazione
Walkthrough: attività

- Fase 1: pianificazione
- Fase 2: lettura
- Fase 3: discussione
- Fase 4: correzione dei difetti
- In ogni fase
 - Documentazione delle attività svolte

Dipartimento di Informatica, Università di Pisa17/22



Verifica e validazione
Inspection: attività

- Fase 1: pianificazione
- Fase 2: definizione lista di controllo
- Fase 3: lettura
- Fase 4: correzione dei difetti
- In ogni fase
 - Documentazione delle attività svolte

Dipartimento di Informatica, Università di Pisa19/22



Verifica e validazione
Inspection: definizione

- Obiettivi**
 - Rivelare la presenza di difetti
 - Eseguire lettura mirata
- Agenti**
 - Verificatori distinti e separati dai programmatori
- Strategia**
 - Focalizzare la ricerca su presupposti
 - Error guessing*

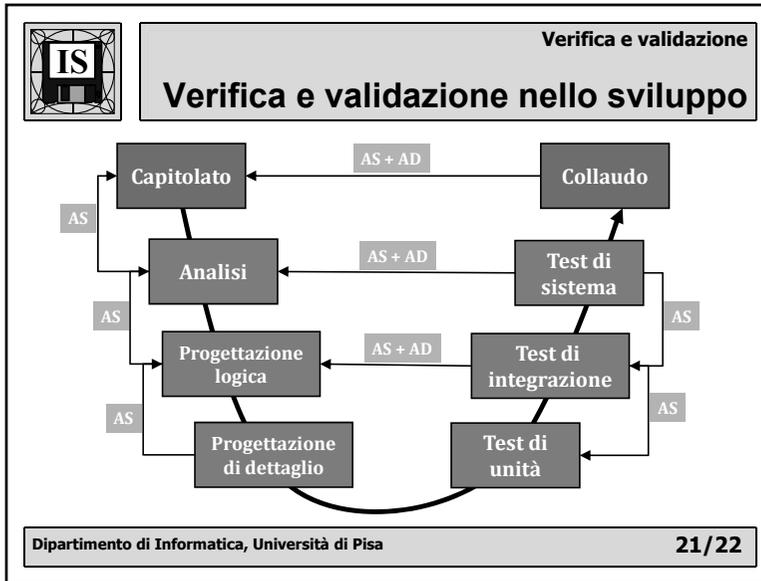
Dipartimento di Informatica, Università di Pisa18/22



Verifica e validazione
Inspection vs. walkthrough

- Affinità**
 - Basati su *desk check*
 - Programmatori e verificatori su fronti opposti
 - Documentazione rigorosa di attività ed esiti
- Differenze**
 - Inspection* basato su (errori) presupposti
 - Walkthrough* richiede maggiore attenzione
 - Walkthrough* più collaborativo
 - Inspection* più rapido

Dipartimento di Informatica, Università di Pisa20/22



Verifica e validazione

Riferimenti

- ❑ **Standard for Software Component Testing, British Computer Society SIGIST, 1997**
- ❑ **M.E. Fagan, Advances in Software Inspection, IEEE Transaction on Software Engineering, luglio 1986**
- ❑ **G.A. Cignoni, P. De Risi, "Il test e la qualità del software", Il Sole 24 Ore, 1998**

Dipartimento di Informatica, Università di Pisa 22/22